Modular Teleoperation Framework

Adhavan Jayabalan, Kritika Iyer, Apoorva Gupta, Namrita Madhusoodanan

***Abstract*—Teleoperation is a growing mode of communication between master and slave devices especially in applications such as minimally invasive surgery. There are a variety of available slaves and masters each having different DH parameters and Degrees of freedom(DOFs). This paper aims at devising a modular teleoperation framework which can merge various different masters and slaves. The framework can take velocities from one robot and feed that to the slave and return a force feedback if necessary to the master. This would be very helpful in surgeries where haptic feedback is valuable.**

***Index Terms*—Teleoperation, framework, Master, Slave, Haptic feedback**

I. Introduction and Background

Teleoperation is a method by which two devices can communicate with each other from remote places[7]. A device that is responsive to another device is termed a slave, and the controlling device is termed a master. This can be useful in various applications, such as in surgeries when a patient is in a place that is remote or a place where the doctor is at risk, or in hazardous situations where people could be exposed to radiation, and in exploring outer space [1]. In 1948, Raymond Goertz designed the first master slave manipulator, while working for the Atomic Energy Commission at Argonne National Laboratory, in order to handle radioactive material[2]. Teleoperation has gained importance and world wide applications in different fields ever since. Teleoperated robots provide a platform to perform tasks like laproscopic surgery,arthroscopic surgery,remote surveillance and exploration of outer space. A lot of astronomical observations have been conducted by telerobotic telescopes. Lunokhod 1 was the first teleoperated robot rover to freely move across the surface of an astronomical object, i.e, the moon. The vehicle could travel a distance of 10km over difficult,rugged terrains, and had the capability to interact with and operate successfully in an unknown, hostile environment[3].

Implementing teleoperation in surgical robots is an upcoming research topic[6]. Such systems have an arm that is controlled by the surgeon and a slave robot that performs the surgery. Sensors on the slave robot are used to give feedback to the surgeon through the master robot using techniques such as haptic/force feedback. In force feedback, forces are sensed at the tip of any instrument of a slave robot using sensors and that force is scaled and reproduced by the master robot[12].

For demonstrating the concept of teleoperation, this project uses 2 devices namely the ABB IRB 120 industrial arm and the Geomagic Touch haptic device. The ABB IRB 120 robot is chosen as the slave, and the geomagic touch haptic device is chosen as the master. The project focuses on moving the ABB robot through teleoperation by a user controlling the geomagic touch haptic device, and sensing the forces experienced by the ABB IRB 120 robot at the master's end.

ABB IRB 120 is a multipurpose industrial arm that has 6 DOFs and is the smallest robot manufactured by ABB weighing just 25kg. The ABB can carry a payload of up to 3kg [9]. The ABB IRB 120 can be mounted on any surface in a vertical or horizontal position as shown in Fig. 1.This robot is designed to function as an industrial arm for performing functions such as welding, painting, soldering etc. There are different choices for the end effector that result in different applications such as grasping and painting. For the purpose of this project, this robot is considered equivalent to a surgical arm, as it can also be mounted with surgery tools like cardiac stabilizers.

Geomagic Touch is a haptic device that applies force feedback on the user's hand. This device is a 6 DOF device having three active joints (J1,J2,J3) and three passive joints(J4,J5,J6) as shown in Fig. 2.Each of the three active joints has a motor, unlike the passive joints. It can be connected to the computer via an on-board Ethernet Port or USB Port.

OpenHaptics is the software used to add the Haptic Device and Haptic Library API Functionality like setting motor torques, getting device state and creating 3D haptic environment using OpenGL[10]. 3D Haptic environment is a way to create virtual objects or shapes. For surgical purposes, these environments can be the shape of organs. Since Geomagic Touch is a haptic device, it lets you feel those virtual objects and if the device is forced to move out of the boundary of those objects, a feedback force can be felt. This functionality of Geomagic Touch to create virtual environments is of great use in surgical applications[8], [5].A surgeon can experience a force feedback that will prevent him from causing unintended damage to organs surrounding the organ being operated.

II. Formal Problem Statement

This paper aims at creating a general or universal framework for teleoperation of various masters with various serial manipulator slaves. This paper will enable a velocity command given from the master to be rippled and mimicked on the slave device where the slave device can give a force feedback to the master if the master has a capability of receiving a haptic signal. The framework should be able to interface any master with any slave effectively and create various modes of input and outputs for example: velocity control and force feedback input and output.

III. Methods

To interface 2 devices with different DOFs or frame settings, it is important to link their movements in such a way that the

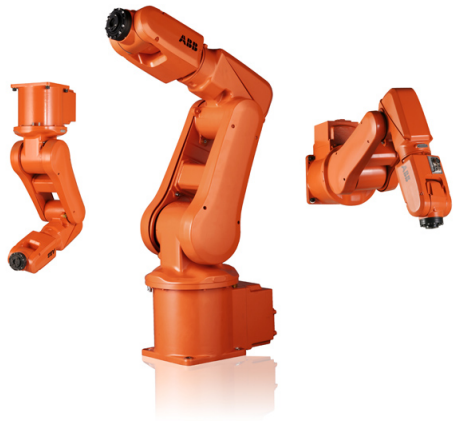


Fig. 1: ABB IRB 120

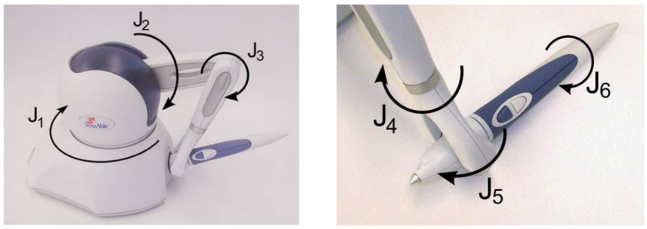


Fig. 2: Geomagic Touch Joints

end effector movement stays the same. To do so, the Homogeneous transformation matrices for each device is required through which the Jacobian matrix can be calculated and the velocity can be controlled. This should be given in ROS nodes and seen in gazebo environment. These steps will be explained in the following sections. The flow of the system can be seen in Fig. 3

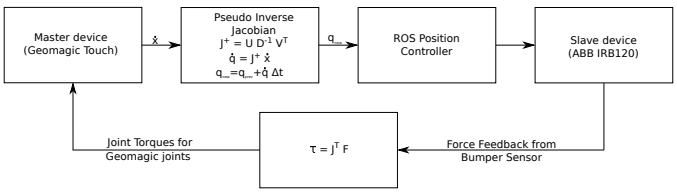


Fig. 3: Project Architecture

A. Environmental setup

The ABB IRB 120 robot is loaded in the gazebo environment. This is used to visualize movements in simulation. The ABB IRB 120 in Gazebo has a bumper sensor attached to it as shown in Fig. 4. This sensor gives an idea of the forces that are acting upon the end effector of the ABB robot. This sensor is simulated using *libgazebo_ros_bumper.so* plugin. Rviz is another visualization software that is used for viewing frames and movement of the robot.

For the interaction of Geomagic Touch device with the computer, OpenHaptics is used. OpenHaptics libraries such as Haptic Device (HD) are included to obtain the functionalities such as Get Joint States, Get Torques etc. The RViz simulation of Geomagic Touch device was established and it was observed that there are only three active joint used to reach any point via the stylus of this device.

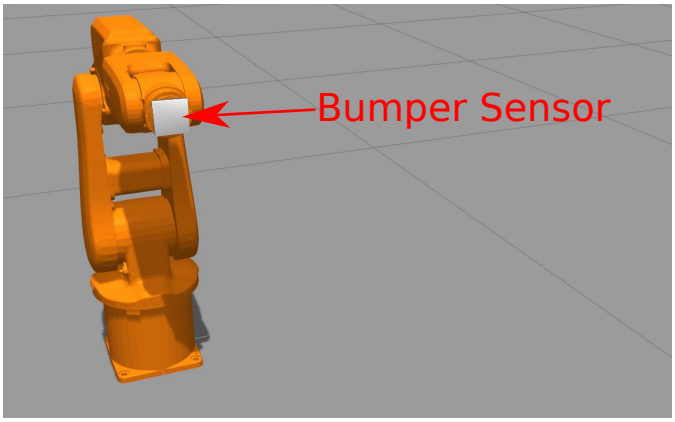


Fig. 4: ABB with Bumper Sensor in Gazebo Environment

B. Modeling the System

The ABB IRB 120 and the Geomagic Touch haptic device that have been used for this project must first be modeled. The devices are analyzed and their frames are set through Denavit-Hartenberg (D-H) convention [11]. The same convention can give a homogeneous transformation matrix from base to tip.

1) Frame Setting for Master and Slave: In Fig. 5, the 6 DOF ABB robot is shown with the assigned frames and the link lengths in the home configuration. In this the Frames 4 and 5 have the same origin. Similarly in Fig. 6 the frames were set according to the D-H convention. In this, frames 2 and 3 are in the same origin and frames 4 and 5 are in the same origin. In both the figures, the blue color frame axis is the z-axis, the red color frame axis is the x-axis and green color represents the y-axis. From these frames the D-H parameters given below were determined.

2) DH parameters: To establish the kinematic model of ABB IRB120 robot, the forward kinematics was established by finding the D-H Parameters as shown in Table I. Similarly the Geomagic Touch's D-H parameters were written as shown in Table II. These Parameters can be taken and placed in Eq. (1) for each frame. Once all the transformations for each frame is obtained the final base to tip transformation (T_6^0) can be obtained.

$$T_i^{i-1} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\alpha_i) & sin(\theta_i)sin(\alpha_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

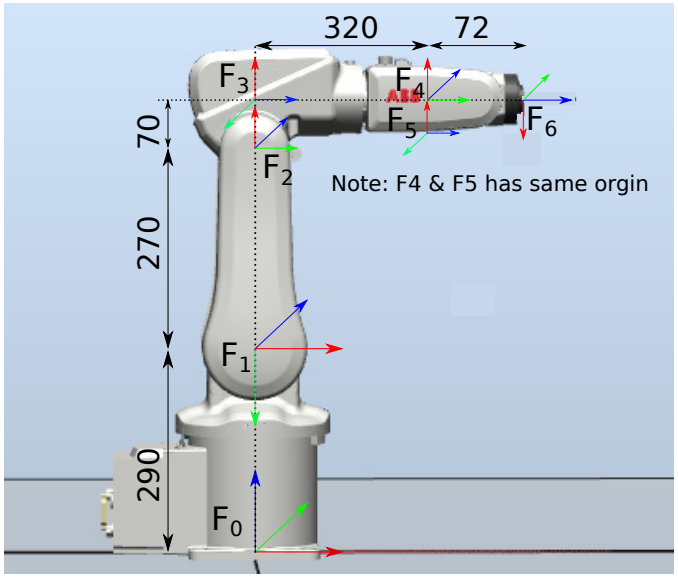
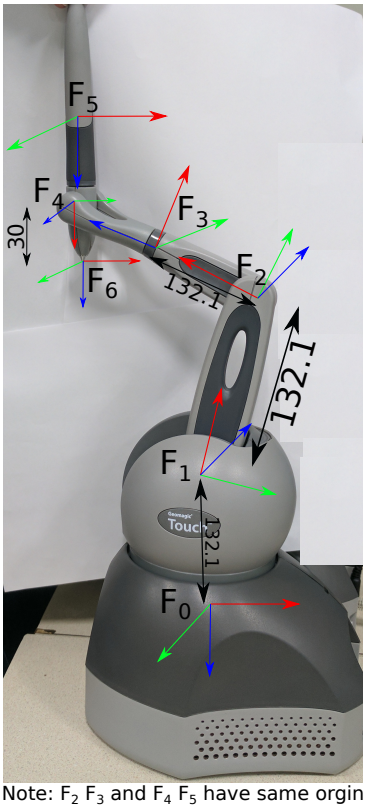


Fig. 5: ABB IRB120 Frame setting at Home Position



Note: F_2 F_3 and F_4 F_5 have same orgin

Fig. 6: Geomagic Frame setting

TABLE I: ABB IRB120 DH Parameters

Frame	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	290	q_1
2	270	0	0	$q_2 - (\pi/2)$
3	70	$-\pi/2$	0	q_3
4	0	$\pi/2$	302	q_4
5	0	$-\pi/2$	0	q_5
6	0	0	72	$q_6 + (\pi)$

TABLE II: Geomagic Touch DH Parameters

Frame	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	-132.1	$q_1 + (\pi)$
2	132.1	0	0	q_2
3	0	$\pi/2$	0	$q_3 + (\pi/2)$
4	0	$\pi/2$	132.1	q_4
5	0	$\pi/2$	0	$q_5 + (\pi/2)$
6	0	0	30	q_6

C. Jacobian

The Jacobian matrix is used to connect the velocities of each joint with the overall velocity of the end effector. The Jacobian is a $6 \times n$ matrix where n is the number of joints. The 6 rows represent $[V_x, V_y, V_z, W_x, W_y, W_z]^T$ where V is velocity and W is the angular velocity.

1) Obtaining the Jacobian: The Jacobian matrix is obtained in 3 parts. First the velocities are calculated and then the angular velocities are calculated. The position vector (column 4) on the (T_6^0) matrix is partially differentiated with each joint variable as shown in Eq. (2). The angular velocities are calculated by taking column 3 of each (T_{i-1}^i) matrix (i.e, the approach vector or the Z axis) and multiplying it with a value η. η is 1 for revolute joints and 0 for prismatic joints. The last 3 rows of the Jacobian can be seen in Eq. (3). The full Jacobian is shown in Eq. (4).

$$J_v = \begin{bmatrix} \frac{\partial T_6^0(1,4)}{\partial q_1} & \frac{\partial T_6^0(1,4)}{\partial q2} & \cdots \\ \frac{\partial T_6^0(2,4)}{\partial q_1} & \frac{\partial T_6^0(2,4)}{\partial q2} & \cdots \\ \frac{\partial T_6^0(3,4)}{\partial q1} & \frac{\partial T_6^0(3,4)}{\partial q2} & \cdots \end{bmatrix} \tag{2}$$

$$J_w = \begin{bmatrix} \eta_1 \times T_{i-1}^i(1:3,3) & \eta_2 \times T_{i-1}^i(1:3,3) \cdots \end{bmatrix} \tag{3}$$

$$J = \begin{bmatrix} \frac{\partial T_6^0(1,4)}{\partial q_1} & \frac{\partial T_6^0(1,4)}{\partial q2} & \cdots \\ \vdots & \vdots & \vdots \\ \eta_1 \times T_{i-1}^i(1:3,3) & \eta_2 \times T_{i-1}^i(1:3,3) & \cdots \end{bmatrix} \tag{4}$$

$$\dot{q} = J^{-1}\dot{X} \tag{5}$$

2) Application of Jacobian: This Jacobian is taken to calculate the joint velocities from end effector velocity over Inverse kinematics method as it is more generic and can be used for all robots if the (T_6^0) matrix is given. Inverse kinematics approach is different for each robot and is generally done manually which makes it infeasible to use in this paper. The joint velocities can be calculated using Eq. (5) where $\dot{q}$ is the vector of joint velocities and $\dot{X}$ is the end effector velocities. The Jacobian is calculated for both master and slave device for the velocity tracking and force feedback applications.

3) Pseudo Inverse Jacobian: The inverse in Eq. (5) ideally should be computed but, the computations are heavy and inverse works only for square matrices. To overcome this, the paper uses the Pseudo-Inverse Jacobian method which give a good approximation of the inverse of the Jacobian found from Eq. (4).

The Pseudo-Inverse Jacobian is obtained by using Singular Value Decomposition(SVD) method where a matrix is separated into 3 different matrices. The matrices then form together to give an approximate inverse called the Pseudo-Inverse denoted by J^+.

By SVD method J can be split into 3 matrices namely S,V,D. J^+ can be obtained by re-ordering the 3 matrices as shown in Eq. (6). Where D^+ is the pseudo inverse of D.

$$J^+ = S \times D^+ \times V^T \tag{6}$$

4) Velocity tracking and Force Feedback: The Jacobian matrix is mainly used to calculate velocities of each joint given an end effector velocity. The calculation can be shown in Eq. (7).

$$\dot{q} = J^+ \dot{X} \tag{7}$$

The Jacobian can also be applied to give details about the forces applied on each joint when a force is applied to end effector. The forces applied to the end effector is expressed in a 6×1 vector holding 3 forces and 3 torques. The torques that each joint experiences can be derived from Eq. (8) where τ is a $n \times 1$ vector where n is the number of joints and F is a 6×1 vector consisting of $[F_x, F_y, F_z, \tau_x, \tau_y, \tau_z]^T$. The F matrix is obtained by taking the output of the master robot force sensors. The values obtained in the τ matrix using the Jacobian of the Slave can then be given to the slave device to give force feedback.

$$\tau = J^T \times F \tag{8}$$

In this way the Jacobians for the ABB and geomagic were obtained.

TABLE III: ABB Joint Limits

Joint #	Limits			
	Position (rad)		Velocity (rad/sec)	Effort
	Min	Max		
1	-2.879	2.879	4.36	20
2	-1.919	1.919		
3	-1.919	1.221		
4	-4	4	5.58	
5	-2.094	2.094		
6	-4	4	7.33	

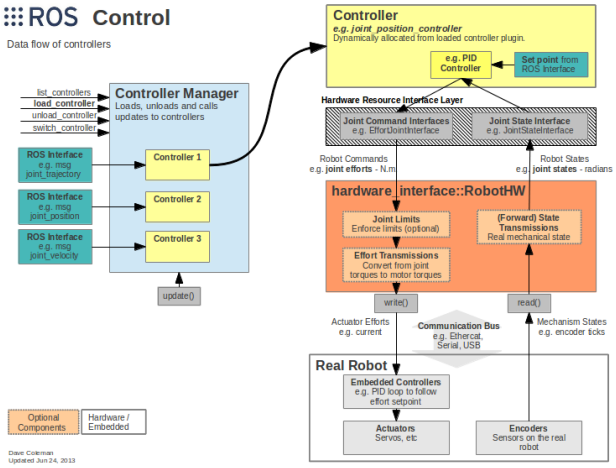


Fig. 7: ROS Control Architecture

D. Controller

To control the joint angles of the ABB IRB120 in Gazebo, ROS control packages are used which have PID controller to track the desired position, velocity or torque for each joint. In this paper, PID position controller has been implemented which can track the desired angular position of each joint. The joint velocities obtained using Eq. (7) are used to calculate the new position using Eq. (9). The obtained position is given to the ROS control manager using the topics *\irb120\joint_i_position_controller\command*.

$$q_i^{new} = q_i^{current} + \dot{q}_i \Delta t \tag{9}$$

The control architecture of the ROS control is shown in Fig. 7. Once the desired joint angles are given to the ROS control, the hardware_interface::RobotHW is used by the controller to know the joint limits (mentioned in Table III) and the controller converges to the desired angular position if it is within the limit.

The performance of the controller to track the desired angular position is totally dependent on the gain values (K_p, K_d, K_i). These values are tuned using rqt_graph tool in ROS. The gain values set for each joint is shown in Table IV.

TABLE IV: PID Gain values

Joint #	K_p	K_i	K_d
1	100	0.01	10
2	100	0.01	10
3	100	0.01	10
4	500	0.1	1
5	300	0.1	2
6	700	0.1	1

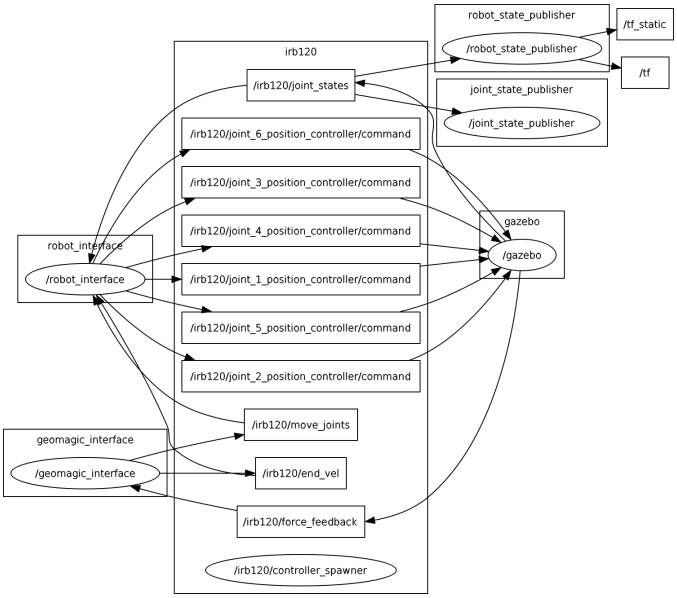


Fig. 8: ROS Nodes & information flow

E. ROS Nodes

The Fig. 8 shows different ROS nodes that are running. The geomagic_interface is the node that interfaces with Geomagic Touch to obtain the end effector velocity and the joint angles. It also sets the torques to the first three active joints of the haptic device. robot_interface node is used to interface with ABB IRB120 to calculate the joint velocities and to communicate with the ROS control package and to the Gazebo plugins.

IV. Results and Discussion

The code for calculating the Forward kinematics and Jacobian was first done on MATLAB and then written in C++. The C++ code included the ROS nodes where the data was transfered via the connections shown in Fig. 8.

A. Velocity and Position Control

To test the output of the code the haptic device was moved with a certain end effector velocity and the same effect was seen in the ABB robot. To obtain this result the movements were scaled up 10 times. The Results were then verified with the MATLAB code written and can be seen in Fig. 10 where the arm is plotted based on the movement in the given time step. **??** shows the various parameters changing as the ABB arm moves. Fig. 12 shows the change in joint velocities of the joints of the ABB. This shows the ABB robot moving in X-axis as given by the Geomagic touch device.

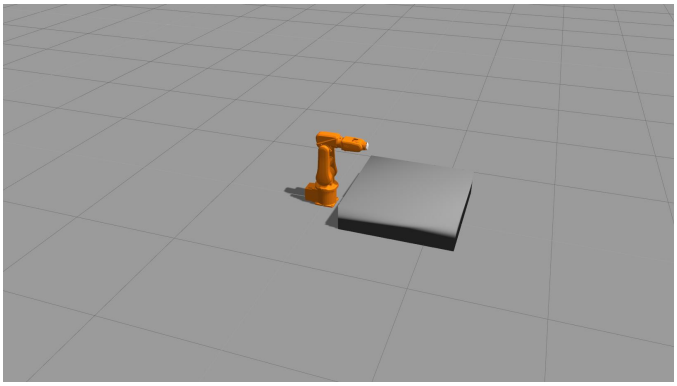


Fig. 9: ABB hitting block for force feedback

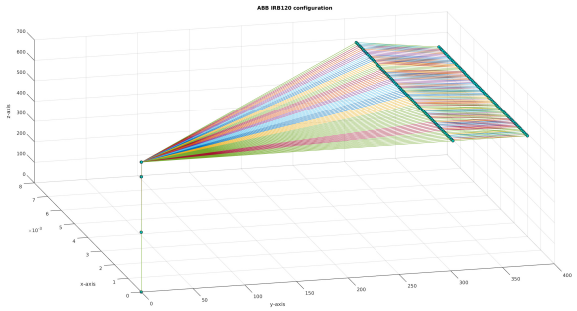


Fig. 10: ABB IRB120 Configuration

B. Force Feedback

The bumper sensor placed on the ABB robot was then tested by moving the ABB robot in the same way and placing a block in the simulation as shown in Fig. 9. The ABB then is moved towards the block. The Bumper sensor senses these forces and gives it to the ROS nodes which calculate the torque to be applied on each motor of the Geomagic Touch to give a force feedback. The forces are scaled up to fit the required force needed. This Torque locks the Geomagic in a position so there is no further movement. This is demonstrated in the video attached with this paper.

C. Challenges Faced

The ROS controller has only a basic PID controller which can efficiently track a set point. The difficulties faced pertaining to this controller was taking time to converge to the given set point within the time step which leads to the inaccuracies in the joint position. This error was propagating as the simulation time was increasing. Another problem faced was with an ABB joint with undesirable behavior which could also have been a result of the PID controller.

V. Future Work

The modular teleoperation framework can be extended using a different hardware setup as well. One such hardware

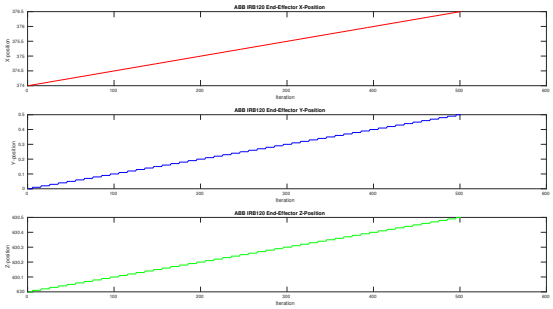


Fig. 11: End Effector Position w.r.t World Frame

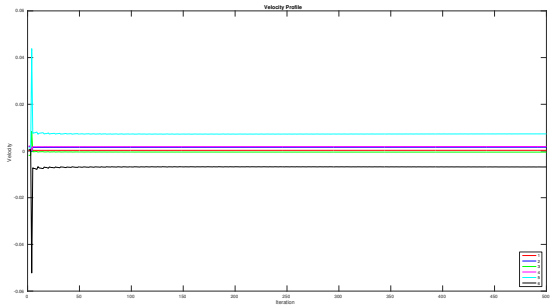


Fig. 12: ABB IRB120 Joint Profile

combination is using the Da Vinci MTM as the master and ABB IRB120 as the slave with the help of custom made Performance Motion Devices (PMD) motor controller.
Teleoperation also has issues like latencies due to limitations in communication speed and computationally heavy calculations. One solution for this problem is using parallel processing. All the computations can be distributed over different processing nodes. Real time control of end effector velocity can be improved using this technique.

VI. Acknowledgement

The Authors would like to thank Prof. Fischer for his continuous support and guidance throughout the project. He was always available and happy to help when needed. The authors are also thankful for the Prof. Zhi Li for allowing us to work on the Geomagic Touch Haptics device in her lab and conduct experiments. The authors would further like to take this opportunity to thank their fellow classmates for their feedback in every step and their aid while facing issues. Last but not the least, thanks goes out to WPI for providing this opportunity to work on such an interesting topic and come up with satisfying results.

References

[1] Preusche, Carsten, Tobias Ortmaier, and Gerd Hirzinger. "Teleoperation concepts in minimal invasive surgery." Control engineering practice 10.11 (2002): 1245-1250.

[2] Goertz, Raymond C. "Mechanical master-slave manipulator." Nucleonics (US) Ceased publication 12 (1954).

[3] Kassel, Simon. Lunokhod-1 Soviet lunar surface vehicle. Vol. 802. No. ARPA. RAND CORP SANTA MONICA CA, 1971.

[4] A lvarez, Ba rbara, et al. An architectural framework for modeling teleoperated service robots. Robotica 24.04 (2006): 411-418

[5] Okamura, Allison M. "Haptic feedback in robot-assisted minimally invasive surgery." Current opinion in urology 19.1 (2009): 102.

[6] Anderson, Robert J. "SMART: A modular architecture for robotics and teleoperation." Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on. IEEE, 1993.

[7] Lichiardopol, S. "A survey on teleoperation." University of Eindhoven, Department Mechanical Engineering Dynamics and Control Group Eindhoven (2007).

[8] Munawar, Adnan, and Gregory Fischer. "A Surgical Robot Teleoperation Framework for Providing Haptic Feedback Incorporating Virtual Environment-Based Guidance." Front. Robot. AI 3: 47. doi: 10.3389/frobt (2016).

[9] https://library.e.abb.com/public/3bd625bab3c7cae1c1257a0800495fac/ROB0149EN_D_LR.pdf

[10] http://www.geomagic.com/files/4013/4851/4367/OpenHaptics_ProgGuide.pdf

[11] Spong, Mark W., Seth Hutchinson, and Mathukumalli Vidyasagar. Robot modeling and control. Vol. 3. New York: wiley, 2006.

[12] Wagner, C., N. Stylopoulos, and R. Howe. "Force feedback in surgery: Analysis of blunt dissection." Proceedings of the 10th symposium on haptic interfaces for virtual environment and teleoperator systems. 2002.

VII. Appendix

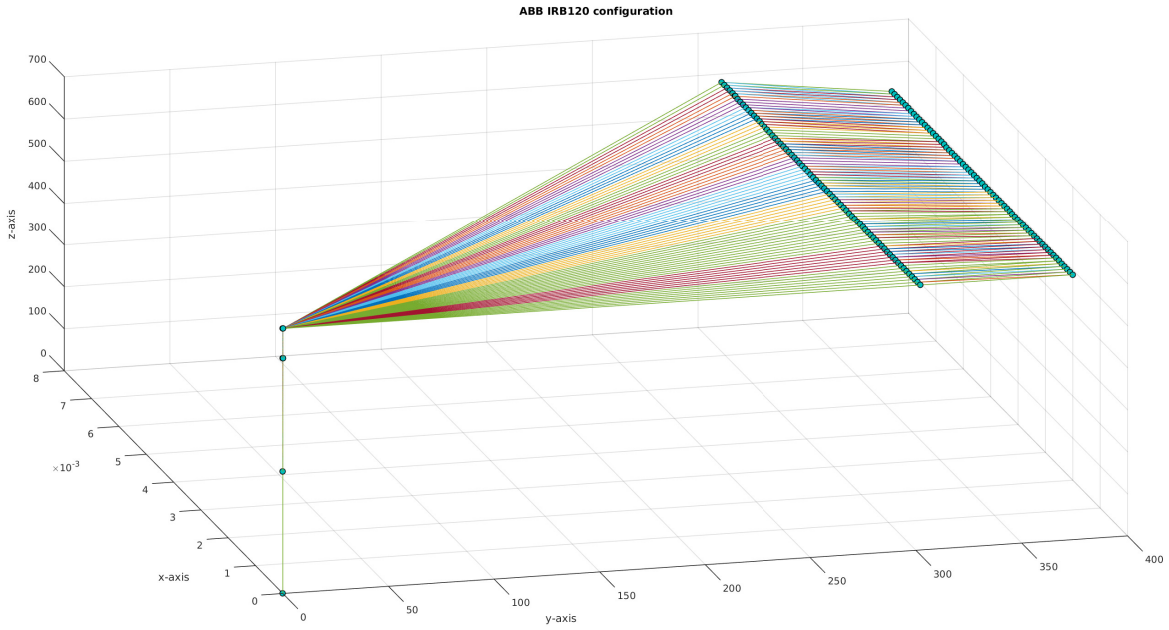
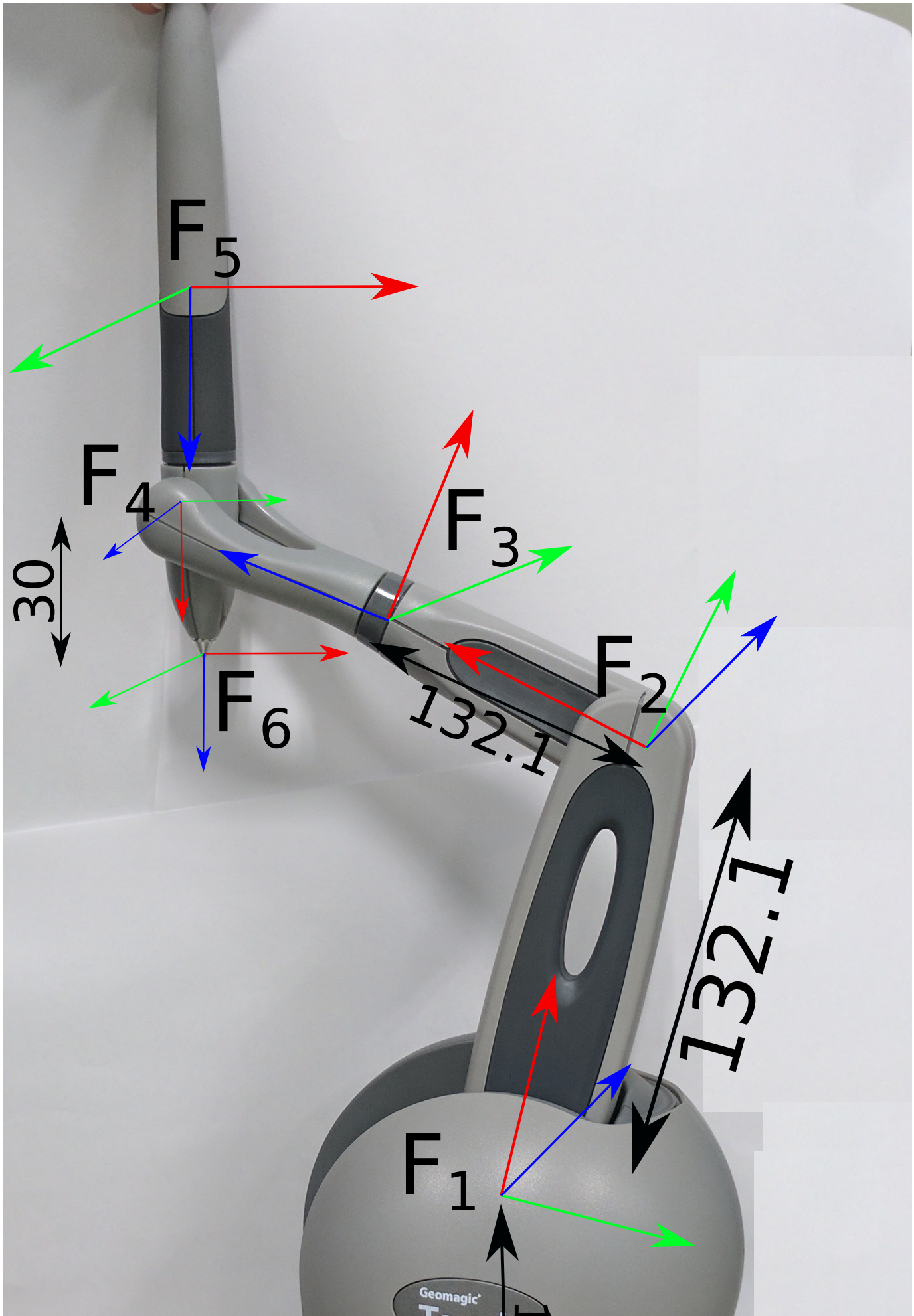


Fig. 13: ABB IRB120 Configuration


F5
F4
F3
F2
F6
F1
30
132.1
132.1
Geomagic

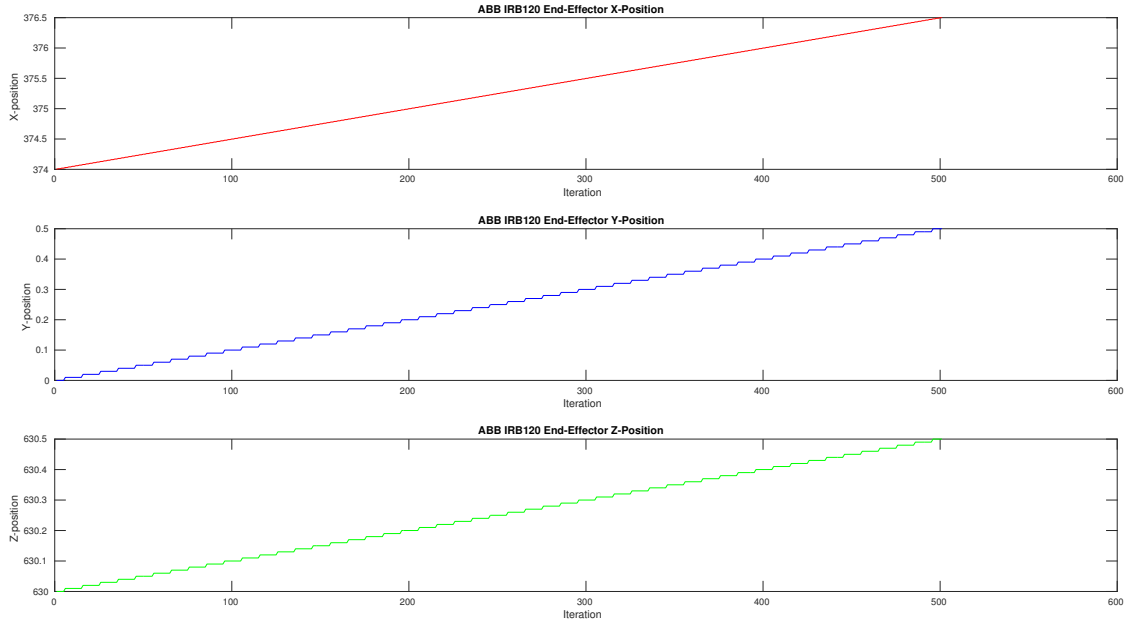


Fig. 15: End Effector Position w.r.t World Frame